

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Systém účtování hovorného pro open-source IP PBX
Billing System for Open-source IP PBX

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Jan Černý**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T059 Mobilní technologie

Téma: **Systém účtování hovorného pro open-source IP PBX**
Billing System for Open-source IP PBX

Zásady pro vypracování:

Jednotlivá volání, ať už jde o odchozí, příchozí či zmeškaná, jsou registrována ve formě generovaných CDR záznamů. Cílem práce je zpracování těchto CDR záznamů a jejich vizualizace v prostředí SW PBX Asterisk.

1. Koncepce pobočkové ústředny Asterisk
2. Struktura CDR záznamů
3. Návrh zpracování CDR záznamů a jejich vizualizace
4. Notifikace při překročení povolených limitů nákladů na volání
5. Zhodnocení výsledků dosažených v praktické realizaci

Seznam doporučené odborné literatury:

VOZŇÁK, M. *Voice over IP*. Ostrava: VŠB - TU Ostrava, 1. vydání, 2008. 176 s.
ISBN 978-80-248-1828-3.

MEGGELN, J., MADSEN, L., KIELHOFNER, K. *Asterisk Cookbook*. O'Reilly Media, 2007. ISBN 978-0596526924.

KOSEK, J. *PHP tvorba interaktivních internetových aplikací*. Praha: Grada, 1998. ISBN 80-7169-737-1.

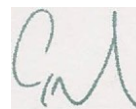
Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: doc. Ing. Miroslav Vozňák, Ph.D.

Datum zadání: 20.11.2009
Datum odevzdání: 06.05.2011



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Poděkování: Chtěl bych především poděkovat panu doc. Ing. Miroslavu Vozňákovi, Ph.D. za neocenitelnou podporu, poskytnutí nápadů a také za odborné vedení mé diplomové práce. Dále bych chtěl poděkovat svým rodičům a přátelům za gramatickou korekturu, psychickou podporu a trpělivost.

Datum odevzdání: 6. května 2011

Diplomant: Bc. Jan Černý

Abstrakt

Diplomová práce se zabývá systémem účtování hovorného pro open-source IP PBX. Práce je zaměřena na open source produkt Asterisk, který ukládá CDR záznamy o hovorech do databáze. Cílem této práce je tyto záznamy přehledně vizualizovat a vypočítat potřebné statistiky. Samotná vizualizace je v práci řešena prostřednictvím webového serveru a PHP skriptů. Druhá část práce pojednává o dalších možnostech využití CDR záznamů. Tato data je možné využít pro detekci možného zneužití VoIP služeb. Proti zneužití VoIP služeb je realizován skript, který v pravidelných intervalech vypočítává statistiky a porovnává s aktuálně nastavenými limity. V případě nadlimitního volání je administrátor serveru informován. Takto realizovaný skript je další možnou obranou proti zneužití VoIP služeb.

Abstract

Graduation thesis is considering of telephone charging system for open-source IP PBX. Thesis targets on an Asterisk open source product, which stores CDR call records into databases. The main goal of this thesis is the right way to prepare a well-arranged visualization and to calculate necessary statistics. The visualization is at this thesis worked out by means of web server and PHP scripts. The second part of thesis focuses on additional possibilities of CDR data assimilation. This information is available for detection misuse VoIP services. There is a script implemented to avoid this VoIP misuse, which periodically predicts statistics and compares currently adjusted limits. The administrator is informed in the case of excess limits by return. Implemented script is another possible prevention against VoIP services misuse.

Klíčová slova

Asterisk, telefonní ústředna, IP telefonie, open source, účtovací systém, SIP, Linux, CDR

Key Words

Asterisk, telephone exchange, IP telephony, open source, billing system, SIP, Linux, CDR

Seznam použitých zkratek a symbolů

ACD – (Automatic Call Distribution) – Automatické distribuované volání
AGI – (Asterisk Gateway Interface) – Rozhraní Asterisku
AMI – (Asterisk manager interface) – Rozhraní Asterisku
CDR – (Call detail records) – záznamy o hovorech v systému Asterisk
H.323 – Signalizační protokol používaný ve VoIP
HTTP – (Hypertext Transfer Protocol) – Internetový protokol používaný pro přenos informací
IAX – (Inter-Asterisk exchange) – Signalizační protokol využívaný Asteriskem
IPsec. – (IP security) – bezpečnostní rozšíření IP protokolu
ISDN – (Integrated Services Digital Network) – Digitální síť integrovaných služeb
IVR – (Interactive Voice Response) – Interaktivní hlasový automat
LAMP – (Linux Apache MySQL PHP) – platforma pro implementaci dynamických webových stránek
MD5 – (Message-Digest algorithm 5) – Hashovací funkce
MGCP – (Media Gateway Control Protocol) – Protokol používaný ve VoIP
NAT – (Network Address Translation) – Překlad síťových adres
PBX – (Private Branch eXchange) – Pobočková telefonní ústředna
PSTN – (Public Switched Telephone Network) – Veřejná telefonní síť
RTP – (Real-time Transport Protocol) – Protokol přenosu v reálném čase
RTCP – (RTP Control Protocol) – Řídící protokol pro RTP
SDP – (Session Description Protocol) – Protokol k řízení multimediálního přenosu pro SIP
SIP – (Session Initiation Protocol) – Signalizační protokol používaný ve VoIP
SQL – (Structured query language) – standardizovaný dotazovací jazyk
SRTP – (Secure RTP) – Zabezpečený protokol přenosu v reálném čase
TLS – (Transport Layer Security) – Protokol poskytující zabezpečenou komunikaci
URI – (Uniform Resource Identifier) – Řetězec znaků pro identifikování či pojmenování zdroje
VoIP – (Voice over Internet Protocol) – Technologie pro přenos digitalizovaného hlasu pomocí paketů v počítačové síti

Obsah

1.	Úvod	1
2.	Softwarová ústředna Asterisk	2
2.1.	Využití Asterisk PBX.....	2
2.2.	Instalace operačního systému Linux	2
2.2.1.	Hardwarové požadavky	3
2.2.2.	Instalace Operačního systému Debian.....	3
2.3.	Instalace Asterisku	3
2.4.	Koncept Asterisku.....	4
2.4.1.	Asterisk Real-time	4
2.4.2.	AGI moduly.....	4
2.4.3.	Asterisk manager API	5
2.5.	Konfigurace Asterisku.....	5
2.5.1.	Signalizační protokol SIP.....	6
2.5.2.	Konfigurace SIP.....	6
2.5.3.	Signalizační protokol IAX	8
2.5.4.	Konfigurace IAX	8
2.5.5.	Číslovací plán	9
2.5.6.	Uvedení nových konfiguračních souborů do provozu.....	10
3.	CDR záznamy	11
3.1.	CDR záznamy v csv formátu	11
3.2.	Nastavení CDR záznamů v systému Asterisk.....	11
3.3.	CDR záznamy v uložené v databázi	12
3.3.1.	MySQL server	12
3.3.2.	HTTP server	12
3.3.3.	Základy databáze.....	13
3.3.4.	Záznam CDR dat do databáze.....	14
3.3.5.	Rozšířené CDR záznamy v databázi	14
3.4.	Aplikace statistiky.....	17
3.4.1.	Poslední hovory.....	17

3.4.2.	Statistiky za období na klienta.....	18
3.4.3.	Statistiky za období na organizační jednotu.....	19
3.4.4.	Výpočet statistik – doba hovoru	19
3.4.5.	Výpočet statistik – cena hovoru	19
3.4.6.	Shrnutí	20
4.	Monitoring hovorů a účtování pro Asterisk pomocí produktů třetích stran	21
4.1.	CDR-stats	21
4.2.	CDRtool.....	21
4.3.	A2billing.....	21
4.4.	Instalace a využití A2billing	22
4.4.1.	Stažení zdrojových souborů	22
4.4.2.	Příprava databáze.....	22
4.4.3.	Vytvoření konfiguračního souboru A2billing.....	23
4.4.4.	Úprava práv a rozšíření konfigurace Asterisku.....	24
4.4.5.	Konfigurace Asterisk Manager	24
4.4.6.	Instalace AGI komponent	24
4.4.7.	Konfigurace pravidelných úloh.....	25
4.4.8.	Instalace webových aplikací	26
4.5.	Konfigurace A2billing	26
4.5.1.	Uživatelské kategorie	26
4.5.2.	Trunky neboli směrování hovoru	27
4.5.3.	Sazby za volání.....	27
4.5.4.	Vytvoření zákazníků.....	28
4.5.5.	Vytvoření SIP zákazníka	28
4.5.6.	CDR záznamy v A2billing	29
4.5.7.	Porovnání volání.....	29
4.5.8.	Denní a měsíční statistiky.....	30
4.5.9.	Jazykové mutace A2billing	31
4.5.10.	Shrnutí A2billing	31
5.	Notifikace při překročení limitů nákladů.....	32

5.1.	Zabezpečení SIP a RTP protokolu	32
5.2.	Pravidelná kontrola CDR záznamů	32
5.2.1.	Bash	33
5.2.2.	Cron	33
5.2.3.	Návrh skriptu	33
5.2.4.	Entita result	34
5.2.5.	Realizace skriptu.....	34
5.3.	Kontrola počtu spojení za minutu	37
5.4.	Další možnosti notifikací	37
5.5.	Notifikace v A2billing.....	38
6.	Závěr.....	39

1. Úvod

Tato práce se zabývá návrhem a realizací systému pro účtování a monitoring hovorného v prostředí Asterisk PBX. V této práci jsou obsaženy informace jakými způsoby Asterisk ukládá záznamy o hovorech a jak je možné s těmito daty nakládat a zobrazovat je v grafické podobě přes webové rozhraní. V práci je zahrnuto seznámení s Asteriskem a nutnou konfigurací pro komunikaci s dalšími systémy. Zahrnuty jsou také informace, jakým způsobem lze Asterisk rozšířit a které rozhraní umožňují rozšíření Asterisku.

Součástí práce je aplikace, která umožňuje přehledně zobrazit informace o hovorech a cenách za služby hovorného. Část práce je věnována návrhu, realizaci a dokumentaci této aplikace. Smyslem této práce je zjistit jakým způsobem lze v ústředně vést vyúčtování za hovory a monitorovat náklady na služby hovorného. V práci jsou shrnuty poznatky, jakým způsobem lze pomocí záznamů o hovorech odhalit zneužití VoIP služeb. Proti zneužití VoIP služeb je realizován skript, který využívá záznamy o hovorech. Na základě těchto záznamů skript vyhodnocuje data a informuje administrátora serveru v případě zneužití VoIP ústředny.

Práce dále pojednává o možnostech tarifkace v systému A2billing, který je komplexním webovým rozhraním pro konfiguraci Asterisku. Tato část práce pojednává, jakým způsobem lze Asterisk propojit s aplikací A2Billing, jakým způsobem lze nakonfigurovat zákazníky a číslovací plán. Blíže jsou rozebrány statistiky, které tato aplikace umožňuje. Tématem této práce je taktéž srovnání jednotlivých softwarových produktů a určení jejich zaměření.

2. Softwarová ústředna Asterisk

2.1. Využití Asterisk PBX

Asterisk je kompletní pobočková VoIP ústředna založená na softwarové bázi. Počátky Asterisku se datují přibližně do roku 1999. Do dnešní doby se Asterisk vyvinul v komplexní softwarovou ústřednu a stal se jedním z nejpoužívanějších open source produktů právě pro pobočkové ústředny. Asterisk dnes poskytuje mnoho možností a služeb pro zpracování hlasu. S výhodou lze v Asterisku implementovat IVR (Interactive Voice Response) s možností distribuovaného automatického volání ACD (Automatic Call Distribution). Asterisk není tedy jen řešení určené pro pobočky, ale může se taktéž pochlubit velmi dobrou rozšiřitelností.

Přídavnými kartami lze rozšířit možnosti Asterisku a využít jej jako telefonní bránu do dalších sítí. Asterisk může spojovat a zpracovávat různé druhy VoIP protokolů jako jsou SIP, MGCP, IAX a H.323 a také mnoho protokolů pro klasické telefonní sítě tak jako jsou POTS a ISDN. Výhoda Asterisku leží v jeho přizpůsobitelnosti, v možnosti implementace různých standardů a jeho otevřeném softwaru.

Asterisk dokáže vytvářet záznamy o volání a efektivně je ukládat do všech možných databázových systémů. Asterisk bohužel nedokáže dále tato data zpracovávat a monitorovat. V Asterisku není žádná z webových aplikací pro monitoring hovorů integrována. K odstranění těchto nedostatků existuje několik open source aplikací, které dokážou komunikovat s Asteriskem a zpřehlednit tak CDR záznamy. Jednou z možností účtovacího systému tedy je vizualizace těchto dat přes HTML. Tyto grafické nadstavby neřeší jen monitoring, ale dovedou si poradit s cenou hovorného a měsíčními fakturami. Dále pak dovedou řešit možnost platby pomocí předplacených služeb nebo měsíční platbu fakturou. Většina těchto nástrojů komunikuje s Asteriskem přes AGI komponenty a dále se pak připojuje do databáze. Jedním z robustních řešení je projekt A2billing, který kromě vyúčtování a monitoringu hovorného řeší i komplexní správu uživatelů a uživatelských účtů. A2billing je řešení připravené pro reálné nasazení, které neřeší jen monitoring hovorů. Aplikace je schopna na základě konfigurace prefixů telefonního čísla směrovat hovory na různé cesty (trunk) s nejlevnější cenou. Velmi efektivně lze takovýmto způsobem ušetřit za volání. Na základě prefixu telefonního čísla se dá v aplikaci rozlišit volání a stanovit tak cenu pro různé předvolby nebo prefixy.

Jednou z dalších možností je využít ukládání CDR dat do databáze a vytvořit aplikaci, která tyto CDR záznamy zpřehlední a vypočítá statistiky hovorného podle zadaných kritérií. Využití této aplikace může být například pro instituce s menším množstvím uživatelů. Aplikace by tedy měla mít funkčnost přehledu nad provolanými jednotkami v reálném čase.

2.2. Instalace operačního systému Linux

Asterisk je dnes hojně podporován ze strany linuxových distribucí. Je možné jej zprovoznit prakticky na jakémkoli systému založeném na Unixu. Starší verze Asterisku byly určeny pro jádra verze 2.4. V dnešní době je však od této starší verze jádra upuštěno a novější

verze Asterisku vyžadují jádro verze 2.6. Asterisk je možné instalovat jak pro malá domácí řešení, tak popřípadě na výkonné servery pro velké firemní zákazníky. Asterisk byl také zkoušen na WRAP deskách a Linksys směrovačích. Pro velmi malý počet uživatelů jsou tyto WRAP zařízení dostačující. Takovéto řešení najde uplatnění v domácnostech nebo v malých firmách.

2.2.1. Hardwarové požadavky

Asterisk potřebuje ovládat aplikace v reálném čase a tak musí přistupovat k procesoru a ostatním hardwarovým zdrojům s prioritním přístupem. V tabulce číslo 1 jsou zaznamenány požadavky a minimální doporučená hardwarová konfigurace v závislosti na počtu kanálů.

Tabulka 1: Systémové požadavky v závislosti na počtu kanálů

Účel nasazení	Počet kanálů	Minimální konfigurace
Domácí uživatel	0 - 5	400 MHz, 256MB RAM
SOHO systém	5 - 10	1GHz, 512MB RAM
Malá korporace	do 25	3GHz, 1GB RAM
Větší korporace	Více než 25	Dual CPU, individuálně

Pro účely této diplomové práce byl vybrán počítač s taktovací frekvencí 1GHz a 512MB RAM.

2.2.2. Instalace Operačního systému Debian

Pro realizaci diplomové práce byla vybrána volně stažitelná distribuce Linuxu Debian verze 6. Tato kapitola vychází z literatury [1] a je pouze doplněním instalační příručky. Instalace operačního systému Debian je intuitivní a je velmi dobře popsána v dokumentaci. Po základní instalaci operačního systému bez grafického rozhraní je třeba doinstalovat balíky pro vzdálenou správu a konfiguraci serveru (Openssh). Následně zkontrolujeme nastavení síťových karet. Doporučuji statické IP adresy nebo přidělování stálé dynamické IP adresy na základě fyzické MAC adresy. Doplnkově doporučuji nastavit firewall.

2.3. Instalace Asterisku

Existují dvě možnosti jak instalovat Asterisk pod systémem Linux.

- První možností je kompilace ze zdrojových souborů a následná instalace. Tento způsob využijeme tehdy, pokud chceme mít instalaci kompletně pod kontrolou.
- Druhou možností je využít balíčků v operačním systému. Debian má velmi propracované řešení závislosti balíků, takže při instalaci nabídne také instalaci souvisejících balíčků a dále pak doporučené balíky. Doporučením v tomto případě je tyto balíky doinstalovat a vyhnout se tak problémům ať už se zvukovým zařízením nebo chybějící výchozí konfigurací. Aktuální verze Asterisku v repositářích je 1.6.

Pro účely této diplomové práce je dostačující instalace z repositářů.

2.4. Koncept Asterisku

Asterisk byl a stále je open source software. Asterisk není tedy uzavřený produkt, který by měl pouze omezené možnosti. Jsou to především doplňky, které dokážou rozšířit působnost Asterisku pro komunikaci například s databází nebo s různými programovacími jazyky.

2.4.1. Asterisk Real-time

Asterisk real-time je modul, který umožňuje Asterisku načítat konfigurace a příkazy z databáze a aplikovat je do číslovacího plánu v reálném čase. Mezi položkami může být zahrnuta například konfigurace uživatelských účtů nebo příkazy číslovacího plánu. Výhodou takového řešení je zejména mechanismus, který okamžitě zahrne změny nastavení účtů bez nutnosti manuálního zásahu do konfiguračních souborů. Další výhodou tohoto modulu může být centrální konfigurace pro více ústředěn založených na Asterisku. Nevýhodou tohoto řešení je zvýšení náročnosti na databázový server, protože Asterisk se dotazuje při každém hovoru databázového serveru. V praxi jsou možné dvě využití tohoto modulu.

- první možností je využít načítání jednotlivých účtů a číslovacího plánu z databáze, změny se projeví ihned po uložení informací v databázi
- druhý způsob je využití pouze doplňujících nastavení, většinou se v praxi využívá pro nastavení cest (trunků) na další poskytovatele a technologie, v tomto případě je nutné pro obnovení konfigurace znovu načíst (reload) konfigurační soubory Asterisku

2.4.2. AGI moduly

AGI (Asterisk Gateway Interface) je rozhraní, které ústřednu neomezuje pouze na vlastní programovací jazyk AEL (Asterisk extensions language). Pomocí AGI komponent může být Asterisk rozšířen o externí aplikace (skripty). Ty mohou být napsány téměř v jakémkoli jazyce. Do těchto skriptů lze nasměrovat jakýkoli datový proud z jádra Asterisku a využít zpracování těchto dat mimo jádro Asterisku. Samozřejmě funguje obousměrná komunikace, takže pokud jádro asterisku vyšle nějaká data do příslušné komponenty, tak může dostat odpověď. Asterisk může pracovat s těmito jazyky.

- zkompileované binární soubory – C, C++ Většinou jde o soubory zkompileované na konkrétní hardware. Manipulace s těmito soubory je poměrně náročná, výhodou ovšem je nenáročnost na hardwarové prostředky tak jako JAVA nebo C#
- výhodou jazyků JAVA a C# je přenositelnost mezi platformami. Nevýhodou je, že tyto jazyky jsou náročnější na systémové prostředky.

- Perl, PHP – interpretované jazyky mají delší odezvu spuštění, protože kód je nejprve zkompileován. Výhodou je malá velikost zdrojového kódu a snadná modifikace.

2.4.3. Asterisk manager API

AMI (Asterisk manager interface) umožňuje připojení do konzole Asterisku prostřednictvím TCP/IP rozhraní. Při komunikaci lze využívat množinu příkazů v rámci konzole. Na toto rozhraní může být přístupováno přes protokol HTTP. Reálné využití této komponenty je především pro monitoring aktuálně připojených uživatelů. Tato data lze získat z konzole a zobrazit je přes webový prohlížeč. Při zadávání příkazů je nutné dodržet formální zápis. Jednotlivé parametry příkazu se oddělují odřádkováním. Konce příkazů jsou pak označeny značkou <CRLF>.

Použitými parametry je určen typ paketu, který může být:

- Akce – požadavek na ústřednu, například využití IAX kanálu
- Odpověď – odpověď ústředny na požadavek
- Událost – informace generované jádrem Asterisku

2.5. Konfigurace Asterisku

Asterisk má ve své základní verzi textové konfigurační soubory umístěné v adresáři /etc/asterisk. Není pravidlem, že Asterisk vždy musí být zkonfigurován přes tyto konfigurační soubory. V dnešní době existují projekty, které řeší konfiguraci Asterisku pouze přes webové rozhraní. Asterisk je rozsáhlý produkt, takže má konfigurační soubory rozdělené do více souborů a to podle zaměření. V hlavním konfiguračním adresáři můžeme najít tyto soubory.

- extensions.conf – číslovací plán (dialplán), podle kterého se směřují hovory
- sip.conf – konfigurace sip serveru, poskytovatelů a uživatelů
- iax.conf – konfigurace protokolu iax, který je využíván ke komunikaci mezi Asterisk servery
- modules.conf – v tomto konfiguračním souboru lze povolit nebo zakázat různé moduly
- rtp.conf – konfigurace přenosu hlasu, pro RTP protokol jsou k dispozici porty 10 000 – 20 000
- manager.conf – konfigurace AMI
- zapata.conf – v tomto souboru jsou k dispozici konfigurace všech rozšiřující telefonních karet
- h323.conf – konfigurace protokolu H.323

Pro přehled jsem uvedl pouze ty nejdůležitější. V adresářové struktuře lze však nalézt velké množství dalších konfiguračních souborů, které umožní z Asterisku udělat kompletní komunikační server s podporou hlasu a chatu.

2.5.1. Signalizační protokol SIP

Protokol SIP vyvinula organizace IETF (Internet Engineering Task Force). Protokol je určen v IP telefonii pro zajištění signalizace. Je mnohem jednodušší než jeho předchůdce H.323 a proto je také častěji používán v IP telefonii. SIP je textový protokol, což napomáhá nejen jednoduchému ladění, ale především je snadno rozšiřitelný. V IP telefonii si nevystačíme pouze s protokolem SIP. Vlastní přenos hovoru zajišťuje protokol RTP. Další důležitý protokol je SDP, který popisuje detaily o vlastnostech zahajovaného přenosu. Tento protokol je přenášen v těle SIP paketů. Podrobnosti o protokolu SIP se můžeme dočíst v literatuře [5].

2.5.2. Konfigurace SIP

Konfigurace SIP kanálu se provádí v konfiguračním souboru `/etc/asterisk/sip.conf`. Možností jak nastavit SIP server v Asterisku je mnoho. V sekci `[general]` je však nutné specifikovat základní parametry SIP serveru jako je port, protokol a další parametry serveru. Pro upřesnění je znázorněn příklad, jak může vypadat základní konfigurace SIP serveru v Asterisku.

[general]

<code>context=sip</code>	<i>; Default context pro příchozí volání</i>
<code>recordhistory=yes</code>	<i>; Povolení ukládání SIP historie</i>
<code>port=5060</code>	<i>; UDP Port (pro SIP 5060)</i>
<code>bindaddr=0.0.0.0</code>	<i>; IP adresa pro navázání spojení (0.0.0.0 pro všechny)</i>
<code>srvlookup=yes</code>	<i>; Povolení DNS server zjištění na odchozí volání</i>
<code>tos=lowdelay</code>	<i>; Nastavení ToS hodnoty (lowdelay,throughput,reliability,mincost,none)</i>
<code>disallow=all</code>	<i>; Zakazání všech kodeků</i>
<code>allow=alaw</code>	<i>; povolení vybraných kodeků</i>
<code>allow=gsm</code>	<i>; další možné kódování G726, G729 na které se vztahují licenční podmínky</i>
<code>allow=ulaw</code>	<i>; standart USA</i>
<code>musicclass=default</code>	<i>; Nastavení hudby při přidržení hovoru</i>
<code>rtptimeout=60</code>	<i>; Ukončení volání když není 60 sec RTP aktivita (volání není ve stavu hold)</i>
<code>rtpholdtimeout=300</code>	<i>; Ukončení hovoru pokud není 300 sec RTP aktivita (volání ve stavu hold)</i>
<code>nat=yes</code>	<i>; NAT nastavení (yes, no, never, route)</i>

Další část je pak konfigurace SIP odchozích spojení. Sekce, která dokáže Asterisk propojit s dalšími SIP servery. Obecně lze napsat příkaz pro spojení s dalším SIP serverem tímto způsobem.

```
register => user[:secret[:authuser]]@host[:port][/extension]
```

Následuje definování uživatelských SIP účtů. Uživatele je možné definovat velmi jednoduchým způsobem. Každý SIP účet může být nadefinován jako typ user, peer nebo friend. Typ user slouží pouze ke komunikaci s příchozím voláním. Nemá tedy práva provádět odchozí volání. Typ friend je účet bez omezení. Typ peer je určen pro konfiguraci poskytovatele. Následující příklad ukazuje, jak může vypadat konfigurace klienta.

```
[5556]  
type=friend  
secret=heslo  
host=dynamic  
context=sip  
disallow=all  
allow=alaw  
allow=gsm  
qualify=yes
```

V Asterisku je možné sledovat dostupnost klientů a to pomocí parametru *qualify=yes*. V případě problémů na SIP kanálu můžeme zjistit zpoždění mezi uživatelem a ústřednou. Tento parametr nám pomáhá ke zkvalitnění monitoringu služeb.

Heslo může být uloženo v textovém formátu nebo může být uloženo pomocí funkce MD5, která nemá inverzní funkci, takže z hash kódu nelze získat zpět heslo.

Pro využití funkce md5 změníme konfigurační soubor následujícím způsobem.

```
auth=md5  
secret=<md5heslo>
```

Heslo můžeme vygenerovat přímo v příkazovém řádku Linuxu a to tímto základním způsobem.

```
# echo -n "uživatel:realm:heslo" | md5sum
```

Standardně je atribut *realm* „asterisk“, pokud jsme tedy v konfiguračním souboru nedefinovali tento atribut jinak. Pro náš konkrétní případ může příkaz vypadat následovně.

```
# echo -n "5556:asterisk:heslo" | md5sum
```


54d2c3b609b0d9f95a37a15d4141787f

Výstupem je hash kód, který vložíme do položky „secret“.

2.5.3. Signalizační protokol IAX

Signalizační protokol IAX (inter asterisk exchange), jak už z názvu vypovídá, je protokol, který slouží ke komunikaci mezi Asterisk servery. Na rozdíl od SIP protokolu dokáže při větším počtu hovorů snížit datový tok RTP paketů. V dnešní době je k dispozici druhá verze tohoto protokolu, která se rozšiřuje i mimo Asterisk. Další výhodou tohoto protokolu je lepší průchod přes NAT především v síti s více Asterisk servery. Pro připojení klienta je však stále v drtivé většině využíván protokol SIP.

2.5.4. Konfigurace IAX

V souboru iax.conf jsou sekce definovány jménem uzavřeným v hranatých závorkách []. Každý soubor iax.conf potřebuje nejméně jednu hlavní sekci [general]. V této sekci je možné definovat nastavení protokolu IAX, tak jako jsou kodeky a vyrovnávání jitteru. Zde je uvedený jednoduchý příklad iax.conf:

```
[general]  
bandwidth=low  
disallow=all  
allow=alow  
allow=ulaw  
jitterbuffer=no  
forcejitterbuffer=no  
tos=lowdelay  
autokill=yes  
register => name:password@server
```

```
[user1]  
type=friend  
host=dynamic  
trunk=yes  
secret=heslo  
context=incoming
```

V sekci [general] potřebujeme příkaz register. Pomocí tohoto příkazu se Asterisk propojí na další server. Mezi servery tedy vytvoříme trunk.

V sekci [user1] se definuje přicházející volání type=friend. Pak se definuje sekce context=incoming, která nám říká, jak se bude přicházející volání chovat v číslovacím plánu. Podstatným parametrem je definování trunku. Další atributy jsou podobné jako při konfiguraci SIP kanálu. Velmi podobně se konfiguruje druhý Asterisk server. Vždy je třeba dát důkladný pozor na kontext, do kterého hovor v číslovacím plánu spadá a zvýšenou pozornost věnovat nastavení hesel. Taktéž je možné využít mechanismu MD5 v případě protokolu IAX.

2.5.5. Číslovací plán

Číslovací plán je centrem celého Asterisku. Veškeré hovory se vždy řídí číslovacím plánem. Podle číslovacího plánu se řídí příchozí a odchozí hovory. Číslovací plán se konfiguruje v souboru extensions.conf. Syntaxe tohoto příkazu je jednoznačně definována.

exten => telefonní číslo nebo znak, priorita, příkaz (parametry)

telefonní číslo nebo znak

Je možné specifikovat absolutně nebo relativně. Absolutní směrování můžeme nastavit u příchozího volání a to pokud máme malé množství telefonních čísel. V tomto případě uvedeme v konfiguraci přímo číslo.

exten => 595111222,1,DIAL(SIP/555)

Taktéž je možné, že číslo odpovídá určitému vzoru. V tom případě je nutné na začátku příkazu použít podtržítka a dále je pak možné použít speciální symboly místo toho, abychom psali čísla. Tímto způsobem lze jednoduše předávat volání na další trunky. Příklad může být následující.

exten => _X.,1,Dial(SIP/\${EXTEN})

V tomto konkrétním případě Asterisk zachytí jakékoliv číslo a bude ho směrovat na SIP kanál. V tomto případě je nutné dát pozor na kontext a vědět tedy kam hovor směřujeme. Znak X v tomto případě označuje čísla od nuly do devíti. Dále je možné využít Z, které značí čísla od jedné do devíti, N značí čísla od dvou do devíti. Pro některé speciální účely využívá Asterisk symboly (s, h, i, t, o). Ty se využívají především při startu různých aplikací na pozadí.

priorita

Prioritu příkazu lze taktéž definovat absolutně pomocí číslic nebo relativně a to pomocí písmen. Na začátku se definuje první priorita číslem jedna. V dalších příkazech se používá znak „n“ který značí následující číslo.

příkaz

Příkaz nebo také spuštění aplikace je posledním krokem uceleného příkazu v číslovacím plánu. Značí, co bude Asterisk provádět. Možností v tomto případě je velmi mnoho.

2.5.6. Uvedení nových konfiguračních souborů do provozu

Pokud konfiguruje Asterisk pomocí textových konfiguračních souborů, nestačí pouze tyto konfigurační soubory editovat. V případě přidání nového hardwaru je nutný restart Asterisku. V případě pouhé změny konfigurační souborů postačí reload. Ten lze provádět standardně přes procesy v operačním systému.

```
#/etc/init.d/asterisk reload
```

Druhou možností je provést znovunačtení konfiguračních souborů prostřednictvím konzole Asterisku a to příkazem reload. Není nutné vždy provádět reload celého Asterisku. Můžeme specifikovat, který kanálový modul se má aktualizovat. V případě změny SIP kanálu, můžeme provést pouze reload SIP kanálu – „sip reload“.

Bližší a podrobnější možnosti konfigurace najdeme především v literatuře [6].

3. CDR záznamy

Základem vyúčtování jsou CDR data (záznamy detailů hovorů). V systému Asterisk je možné ukládat data v textovém formátu nebo přímo do databáze.

3.1. CDR záznamy v csv formátu

Pokud jde o ukládání do souboru, tak je o každém hovoru veden záznam v csv souboru, který se standardně ukládá do `/var/log/asterisk/cdr-csv/Master.csv`. Soubor dat můžeme specifikovat na uživatele (per user) nebo na konkrétní zařízení (SIP, zaptel).

Pro ukázkou vybírám nejdůležitější parametry:

accountcode: podle tohoto parametru se určuje sazba za hovor

src: telefonní číslo volajícího

dst: telefonní číslo volaného

clid: uživatelská identifikace volaného

start: přesný čas a datum hovoru

answer: čas vyzvednutí hovoru

end: čas konce hovoru

duration: celkový systémový čas v sekundách od doby vytočení čísla k ukončení hovoru

billsec: celkový čas hovoru od vyzvednutí k ukončení, toto je čas, který se účtuje

disposition: záznam o hovoru, jestli byl uskutečněn nebo jestli bylo obsazeno atd.

3.2. Nastavení CDR záznamů v systému Asterisk

Standardně jsou CDR záznamy deaktivovány, proto je nutné je nejprve aktivovat. Přihlásíme se do konzole Asterisku a zkontrolujeme přítomnost modulu `cdr_csv.so` pomocí příkazu „`module show like cdr`“. Pokud modul `cdr_csv.so` nevidíme ve výpisu modulů, tak provedeme aktivaci modulu `cdr_csv.so` a to pomocí příkazu „`module load cdr_csv.so`“, který zadáme do konzole Asterisku.

Lépe je tuto změnu zaznamenat do konfiguračního souboru, tak aby po restartu serveru byla změna zahrnuta do konfigurace. Je tedy nutné editovat konfigurační soubor `/etc/asterisk/modules.conf`. V tomto souboru dopíšeme řádek „`load cdr_csv.so`“ a restartujeme asterisk. Výsledný soubor se záznamy o hovorech pak může vypadat následovně:

```
"","5556","5555","sip","""5556""<5556>","SIP/5556-081d4d60","SIP/5555-081de2d8","Dial","SIP/5555","2010-06-29 09:56:49","2010-06-29 09:56:54","2010-06-29 09:57:09",20,15,"ANSWERED","DOCUMENTATION","asterisk-1277805409.1",""
```

Soubor obsahuje jeden záznam o provedeném hovoru z telefonního čísla 5556 na telefonní číslo 5555. Hovor byl proveden na kanálu SIP, na hovor druhá strana odpověděla a hovor trval 15 sekund.

Další zpracování těchto dat by bylo možné pomocí programu AWK a to na úrovni operačního systému Linux. Toto zpracování by přicházelo v úvahu pouze v případě exportu do souboru bez podpory databáze. Využití takovýchto dat uložených v textových souborech pro účely statistik je nevyhovující. Takto uložená data mají význam především pro logování.

3.3. CDR záznamy v uložené v databázi

Další možností kam ukládat CDR záznamy jsou různé typy databází. Asterisk podporuje všechny možné druhy databázových systémů. PostgreSQL, SQLite, MySQL jsou jedny z mála běžně používaných. Asterisk má v sobě i modul ODBC, takže není problém data ukládat přes ODBC například i do komerčních databázových systémů (Oracle, MSSQL).

3.3.1. MySQL server

MySQL je relační databázový systém. Za jeho vznikem stála firma MySQL AB, která nyní patří do portfolia Sun Microsystems. MySQL je multiplatformní databázový server, který je možný provozovat na různých operačních systémech. Nejčastěji je však nasazován v sadě s dalším nekomerčním softwarem v tzv. kombinaci LAMP.

V diplomové práci je využit databázový systém MySQL, který pak dále slouží pro projekt A2Billing. Asterisk přesně nespecifikuje požadavky na verzi databázového serveru. V našem konkrétním případě využijeme jednu z posledních verzí databázového serveru a to verzi 5.1. Další kroky předpokládají nainstalovaný a nakonfigurovaný databázový server a to podle literatury [2].

3.3.2. HTTP server

HTTP server je program, který na základě požadavků klienta, formulovaný protokolem http/https, zasílá požadovaný obsah dat nebo HTML dokument. Data musí být uložena v místě, které je serveru přístupné.

Rozlišují se dva základní typy www dokumentů, které jsou zasílány uživateli.

- statické dokumenty – označovaný obsah je předem pevně daný
- dynamické dokumenty – obsah daného dokumentu je generovaný na základě různých podmínek skriptovacího jazyka, který dokument sestavuje.

Apache

Je http open source http server. V současné době je jedním z nejpoužívanějších na platformě Linux. Apache v této diplomové práci neřeší pouze odpovědi na požadavky klientů,

ale řeší také zabezpečení aplikace pomocí SSL a autentizaci uživatelským jménem a heslem. Instalace a konfigurace je popsána v dokumentaci projektu. [4]

3.3.3. Základy databáze

Při návrhu databáze se využívají pojmy entita, atribut a relace. Jsou to základní jednotky při návrhu logického schématu databáze [3].

Entity

Pomocí entit jsou při návrhu modelu popisovány určité objekty, o kterých se shromažďují informace. Jedná se o třídu, která definuje určité jedinečné vlastnosti objektu. Entity jsou navrhovány tak, aby dvě entity popisující různé objekty neobsahovaly duplicitní vlastnosti, které mají společné.

Atributy

Reprezentují skutečnou jednotku. Entity zahrnují atributy, pomocí kterých jsou popsány charakteristické vlastnosti. Například bude-li entita vyjadřovat mobil, tak zjednodušeně může obsahovat atributy:

- značka
- model
- GSM pásma
- barva

Relace

Relace vyjadřují vztahy mezi souvisejícími entitami. Relace mezi entitami mohou být omezeny kardinalitou. Ta určuje maximální počet instancí entit jednotlivých stran dané relace. Relace mohou být typu:

- jedna k jedné $\{(1:1)\}$;
- jedna k více $\{(1:N)\}$;
- více k více $\{(N:M)\}$;
- rekurzivní relace.

Datové typy

Datovými typy jsou blíže specifikovány atributy. Pomocí datových typů lze omezit určitou množinu informací na znaky, které jsou pro konkrétní atributy smysluplné. To napomáhá databázovému systému, jakým způsobem bude data zpracovávat a uchovávat. Například je-li použit dvakrát datový typ Integer, databázový systém ví, jak s těmito daty naložit, pokud bude požadováno jejich sečtení, násobení. Při

jiném datovém typu by se mohl výsledek operace lišit od očekávaného. Datové typy používané v databázích jsou do jisté míry standardizovány, ale každý výrobce používá i mnoho vlastních datových typů. Kolekce datových typů jsou předem stanoveny konkrétním databázovým systémem.

3.3.4. Záznam CDR dat do databáze

Asterisk v základní konfiguraci neukládá žádné CDR záznamy. Pro ukládání CDR záznamů do databáze je třeba doinstalovat příslušný modul a modifikovat konfiguraci Asterisku. Modul má název asterisk-mysql a je to komplexní balík, který umožňuje ukládat CDR záznamy do databáze, ale také umožňuje z databáze načítat a měnit konfigurace za běhu systému.

Modul aktivujeme v konfiguračním souboru `/etc/asterisk/modules.conf`, kde přidáme řádek `load => cdr_addon_mysql.so`. Vytvoříme konfigurační soubor tak, aby bylo možné ukládat CDR záznamy do databáze. Konfigurační soubor má název `cdr_mysql.conf` a jeho obsah vypadá následovně.

```
[global]
hostname=localhost
user=uživatel pro připojení k databázi
password=heslo
dbname=jméno databáze
table=entita
```

3.3.5. Rozšířené CDR záznamy v databázi

Asterisk poskytuje pouze základní atributy, které mají především význam pro vyúčtování. Tyto atributy lze rozšířit o další, které lze s výhodou použít pro monitoring. Data pro monitoring jsou především IP adresy volajícího a údaje z hlavičky SIP paketů. Tato data budou vhodná především pro veškeré odhalování podvodů.

Samozřejmostí je mít tyto informace v logu a v případě větších problémů se v tomto logu dají dohledat. Pokud však máme tu možnost, není problém si některé údaje uložit do databáze a dále s nimi pracovat.

Je nutné rozšířit číslovací plán, konkrétně soubor `extensions.conf` o následující řádky.

```
exten => h,l,noop("extended CDR")
exten => h,n,set(CDR(hangupcause)=${HANGUPCAUSE})
exten => h,n,set(CDR(peerip)=${CHANNEL(peerip)})
exten => h,n,set(CDR(recvip)=${CHANNEL(recvip)})
exten => h,n,set(CDR(fromhead)=${CHANNEL(from)})
```

```

exten => h,n,set(CDR(uri)=$${CHANNEL(uri)})
exten => h,n,set(CDR(useragent)=$${CHANNEL(useragent)})
exten => h,n,set(CDR(codec1)=$${CHANNEL(audioreadformat)})
exten => h,n,set(CDR(codec2)=$${CHANNEL(audiowriteformat)})
exten => h,n,set(CDR(llp)=$${CHANNEL(rtpqos,audio,local_lostpackets)})
exten => h,n,set(CDR(rlp)=$${CHANNEL(rtpqos,audio,remote_lostpackets)})
exten => h,n,set(CDR(ljitt)=$${CHANNEL(rtpqos,audio,local_jitter)})
exten => h,n,set(CDR(rjitt)=$${CHANNEL(rtpqos,audio,remote_jitter)})

```

Tyto řádky je nutné vložit do konkrétního kontextu v číslovacím plánu. Pro každý parametr je nutné mít příslušný atribut v databázi. Většina těchto parametrů využívá kanálovou funkci (CHANNEL). Pomocí této funkce lze monitorovat kompletní průběh hovoru. Výstupem této funkce může být například kódování využívané během hovoru, informace o účastnickém zařízení. Dále je to pak kvalita spojení, kterou můžeme zjistit z počtu ztracených RTP paketů a jitteru. Možností jak využít tuto funkci je několik. Pro monitoring bohatě postačí interní a veřejná IP adresa volajícího a potom informace s hlavičky SIP paketu. Takovýmto jednoduchým způsobem můžeme rozšířit CDR záznamy v databázi.

Entita CDR

Záznamy o volání jsou nyní poskytovány Asteriskem, před samotným ukládáním záznamů je však nutné aby databáze obsahovala entitu CDR. Formát entity je zobrazen v tabulce 1.0

Tab 1.0

Atribut	Datový typ
calldate	datetime
clid	varchar(80)
src	varchar(80)
dst	varchar(80)
dcontext	varchar(80)
channel	varchar(80)
dstchannel	varchar(80)
lastapp	varchar(80)
lastdata	varchar(80)
duration	int(11)
billsec	int(11)
disposition	varchar(45)
amaflags	int(11)
accountcode	varchar(20)
userfield	varchar(255)
uniqueid	varchar(32)

peerip	varchar(80)
recvip	varchar(80)
fromhead	varchar(80)
uri	varchar(80)
useragent	varchar(80)

- calldate – časová značka, čas pokusu o spojení
- clid – identifikační hodnota volajícího
- src – účet volajícího
- dst – číslo volaného
- dcontext – kontext volaného
- channel – obsahuje informaci o použitém příchozím kanále
- dstchannel – obsahuje informace o použitém odchozím kanále
- lastapp – poslední příkaz asterisku před ukončením hovoru
- duration – celkový využitý čas Asterisku. započtený od vytočení volaného do doby ukončení hovoru
- billsec – účtovací čas, toto je přesná hodnota délky hovoru
- disposition – informace o průběhu hovoru. ANSWERED, NOANSWER, BUSY, FAILED - odpovězeno, nezastiženo, obsazeno, chyba hovoru
- amaflags – hovory lze označit různými parametry - billing, documentation, omit
- accountcode – proměnná podle které lze specifikovat tarif a určit tak cenu hovoru
- uniqueid – Asterisk je schopen po úpravě generovat jedinečné identifikace hovoru
- userfield – volitelné pole
- peerip – interní IP adresa volajícího
- recvip – externí IP adresa volajícího, pokud uživatel není za NAT, jsou tyto dvě IP adresy totožné
- fromhead – údaj z hlavičky SIP paketu „from“
- uri – aktuální adresát požadavku

Entita Tarif

Entita tarif uchovává informace o ceně za minutu volání. Cena je po celou dobu hovoru lineární. Podle této entity lze vypočítat celkovou cenu hovoru.

tab.1.1 Entita tarif

Atribut	Datový typ
accountcode	varchar(60)
price	int(11)

- accountcode – podle tohoto atributu se porovnává cena hovoru

- price – tento atribut určuje cenu za minutu volání

Entita Organization

Tato entita uchovává informace o příslušnosti klientů k organizačním jednotkám. Každý uživatel patří právě do jedné organizační jednotky, ale může patřit i do více organizačních jednotek, toto ale v reálném prostředí není běžné.

entita Organization

Atribut	Datový typ
orgid	varchar(32)
clid	varchar(32)

- orgid – je id organizační jednotky
- clid – je identifikace uživatele

3.4. Aplikace statistiky

Účelem této aplikace je především zobrazit informace o posledních hovorech. Vypočítat dobu hovoru za zadaný čas a stanovit celkovou cenu. Tato kapitola slouží zároveň jako uživatelský manuál k webovému rozhraní aplikace.

3.4.1. Poslední hovory

Pod touto položkou se naskytá možnost zobrazit výpis posledních hovorů. Základem je přehledné zobrazení času volání, telefonních čísel, která se účastnila hovoru, účtovaná doba. Tato položka slouží především pro náhled na CDR záznamy. Má tedy povahu přehledného logu. Dále je možné zjistit IP adresy volaného. Ve sloupci „peerip“ je interní IP adresa volajícího. Ve sloupci „recvip“ je veřejná IP adresa volajícího. Dále je pak možné vidět sloupec „from“. Informace v tomto sloupci jsou zjištěny s hlavičky SIP paketu.

V tomto seznamu je možné listovat v případě, že výsledků v databázi je více než dvacet. Listování zobrazuje přesný počet stran. Na každé straně se zobrazí maximálně 20 záznamů.

Asterisk statistiky								
Posledni hovory Statistika klientu za obdobi Statistika organizacnich jednotek za obdobi								
Posledni hovory								
Strana: 1/2 >>								
Cislo volajiciho	Datum	Typ	Cislo volaneho	Delka spojeni	Uctovana doba	peerip	recvip	from
5555	2011-05-02 14:38:13	ANSWERED	4441	0:00:16	0:00:13	192.168.10.12	192.168.10.12	sip:5555@192.168.10.6
4441	2011-05-02 14:37:32	ANSWERED	4440	0:00:12	0:00:09	192.168.10.22	192.168.10.22	sip:4441@192.168.10.6
4440	2011-05-02 14:37:08	ANSWERED	4441	0:00:16	0:00:11	192.168.10.12	192.168.10.12	sip:4440@192.168.10.6
5556	2011-04-28 19:52:41	ANSWERED	5555	0:00:08	0:00:05	192.168.10.12	192.168.10.12	sip:5556@192.168.10.6
5556	2011-04-28 19:52:22	NO ANSWER	5556	0:00:06	0:00:00	192.168.10.12	192.168.10.12	sip:5556@192.168.10.6

3.4.2. Statistiky za období na klienta

Pod touto položkou se naskýtá možnost výpočtu celkové účtovací doby a ceny hovorného. Pro správný výpočet je nejprve nutné zadat parametry výběru do zobrazeného formuláře. Formulář má celkem 4 kategorie výběru parametrů hledání.

Číslo zákazníka - Identifikace volajícího

Pod tímto výběrem se naskýtá možnost zvolit konkrétní číslo zákazníka nebo vybrat všechny pro přehledné statistiky.

Asterisk statistiky

[Poslední hovory](#) [Statistiky za období](#)

Cislo zakaznika

☐ Vsiichni klienti

Obdobi

☐ Rok

☐ Mesic

☐ Den

Typ

▼

Volane cislo

☒ Jakokoliv

Zobrazit

Parametry vyhledavani

Cislo klienta:5555, zacatek obdobi: 22 Apr 2011 00:00:00, konec obdobi: 22 Apr 2011 23:59:59, typ:ALL

Celkove statistiky za obdobi

Celkova uctovana doba hovoru : 0:00:09
Celkova uctovana cena : 0.7500

Vypis hovoru

Cislo	Datum	Typ	Cislo volaneho	Delka spojeni	Uctovana doba
5555	2011-04-22 08:24:30	NO ANSWER	5556	0:00:21	0:00:00
5555	2011-04-22 08:33:50	BUSY	5556	0:00:01	0:00:00
5555	2011-04-22 08:33:58	ANSWERED	5556	0:00:11	0:00:09

Obdobi

V této sekci je možné specifikovat, jestli se má vyhledat pouze podle roku nebo podle měsíce a roku nebo zahrnout všechny tři časové údaje (rok, měsíc, den). Chceme-li například zobrazit všechny hovory za rok 2011 a měsíc duben, tak zatrhneme položku měsíc a vyplníme příslušný rok a měsíc. Chceme-li vybrat konkrétní den, tak je nutné vyplnit všechny tři parametry (Rok, Měsíc, Den) a zatrhnout položku den.

Typ

Je možné si zobrazit hovory, které byly zodpovězeny, bylo obsazeno, nebo druhá strana neodpověděla.

Volané číslo

V této sekci lze specifikovat číslo volaného. Pro výběr jsou stejné možnosti jako u čísla volajícího. Můžeme specifikovat konkrétní číslo nebo vybrat všechna čísla.

3.4.3. Statistiky za období na organizační jednotku

Ve firemní sféře dnes převažuje názor rozdělení do menších celků. Tyto menší celky můžeme označit jako středisko, organizační jednotka nebo útvar. Možností označení je několik. Korporace může mít přehled nad jednotlivcem nebo může mít přehled nad konkrétním útvarem. Princip výběru parametrů je stejný jako u statistiky za období na klienta. Místo číslo klienta je nutné zadat číslo organizační jednotky. V našem případě jsou definovány dvě organizační jednotky s čísly 10 a 11.

Organizační jednotka 10 sdružuje dva účastníky s telefonními čísly 5555 a 5556. Druhá jednotka potom čísla 4440 a 4441.

3.4.4. Výpočet statistik – doba hovoru

Aplikace vypočítává celkovou účtovací dobu podle atributu „billsec“ u každého hovoru. V závislosti na zvolených parametrech lze získat několik specifických přehledů a několik specifických dob hovoru a to nejen na základě časových údajů. Aplikace nabízí možnosti získání komplexních ročních, měsíčních a denních statistik. Na základě těchto statistik lze sledovat a porovnávat zvýšení či snížení počtu hovorů.

Taktéž může nastat případ, kdy některý z uživatelů korporace vyřizuje v pracovní době své soukromé hovory s partnerem (manžel, manželka). Tyto hovory mohou mít za následek vysoký účet za telefonní služby. Hovory lze filtrovat na základě čísla volaného a lze u nich zobrazit dobu a cenu.

přepočet sekund na hodiny

Asterisk ukládá čas v sekundách. Pro přehlednost je třeba mít tento čas zobrazen v hodinách, popřípadě minutách. Pro přepočet času program využívá funkci sec2hms, která byla převzata ze stránek <http://www.laughing-buddha.net/php/lib/sec2hms>. Tato funkce je napsána v programovacím jazyce PHP a zapadá do konceptu aplikace. Přepočet času se provádí až po sečtení všech výsledných časů.

3.4.5. Výpočet statistik – cena hovoru

Asterisk dokáže u každého hovoru specifikovat atribut „accountcode“. Na základě toho atributu jsou v tabulce tarif stanoveny ceny pro konkrétní kód. Tento atribut můžeme nastavit

ještě před započítáním hovoru a to v konfiguraci číslovacího plánu na základě předvolby nebo kontextu. Příklad konfigurace:

exten => s,l,Set(CDR(accountcode)=3)

Na základě toho atributu je vypočítaná cena všech zvolených hovorů.

3.4.6. Shrnutí

Aplikace statistiky najde uplatnění především jako doplněk všech pobočkových ústředen. Majitelé těchto ústředen dostávají v pravidelných intervalech účet za telefonní služby od poskytovatele služeb, ve většině případů tak firma nemá přehled online. Pomocí aplikace statistiky si může zobrazit online data. Výhodu aplikace vidím především v menších pobočkových ústřednách, kde se CDR záznamy ukládají do textových souborů. Takováto řešení by se mohla rozšířit o webovou aplikaci. Další výhodou aplikace je ukládání IP adres, které může být využito k dohledání volajícího v případě zneužití VoIP služeb.

4. Monitoring hovorů a účtování pro Asterisk pomocí produktů třetích stran

4.1. CDR-stats

Tento produkt je výrobkem firmy Star2Billing. Tato firma se v posledních letech velmi angažuje do vývoje CDR softwaru pro Asterisk a jiné softwarové ústředny. Program je založen na programovacím jazyku Python. Je to program s webovým rozhraním a databází, do které se ukládají CDR záznamy. Jako databázový server je možné využít již zmiňovaný MySQL server, Postgresql, Sqlite3 nebo komerční Oracle. Aplikace využívá knihoven Django. Tyto knihovny jsou rozšířením základní instalace jazyka Python a to především v oblasti webových aplikací. Django je určen především pro programátory, kteří vyvíjí webové aplikace, ve kterých se opakuje několik základních kódů. Tento problém řeší framework, což jsou knihovny usnadňující práci. Projekt je poskytován pod licencí AGPL, což je třetí verze licencování GPL. Jsou tedy k dispozici i zdrojové kódy.

CDR-stat je přehledná webová aplikace, která umožňuje online náhled na CDR záznamy. Tyto záznamy dokáže přehledně zobrazit. Velkou výhodou tohoto projektu je grafická přehlednost a to díky projektu Django. CDR-stat má velké množství grafů a přehledné statistiky. Nevýhodou tohoto softwaru je, že neřeší tarifkaci. Tento produkt řeší pouze přehledné statistiky, které dokáže zobrazit do grafu.

Jeho využití vidím především na méně náročných VoIP ústřednách a to pouze pro náhled CDR záznamů a vytíženost VoIP ústředny. Dále neukládá IP adresy účastníků hovor, takže následné zjišťování útoku v případě zneužití VoIP služeb je problematické.

4.2. CDRtool

Je také software založený na open source řešení. Tento produkt byl původně určen pro různé VoIP ústředny. V posledních letech se však vyhranil a je zaměřen především na VoIP ústřednu založenou na OpenSIPS. Neexistuje tedy dokumentace, která by popsala, jakým způsobem lze CDRtool propojit s Asteriskem. Projekt umožňuje tarifkaci a nastavení limitů pro volání, čímž spadá do vyšší kategorie. Samozřejmostí je přehledný náhled na CDR záznamy, třídění a výběr podle konkrétního data.

4.3. A2billing

V současnosti je možné využít nepřeberné množství aplikací pro účtování CDR záznamů. Jedním s nejběžnějším a nejpoužívanějším softwarem je A2Billing. Program nabízí

velké množství funkcí. Jedním ze základních je správa uživatelů přes webové rozhraní. A2Billing nabízí informace v reálném čase o vyúčtování. Tento systém je připraven jak na platbu formou předplacených karet, tak na formu měsíčních faktur. Dále jsou k dispozici statistiky a monitoring celkového zatížení ústředny. Toto je jen malý výčet, co aplikace A2Billing umožňuje. Aplikace je vyvíjena pod licencí AGPL. Dokumentace a popis všech modulů aplikace není k dispozici online. Respektive aplikace je produktem firmy Star2billing, která již zde byla zmíněna. Firma se snaží vydělat peníze na další vývoj pomocí implementace tohoto projektu ve firemním prostředí. Nabízí placenou podporu, takže veškerou dokumentaci si chrání pouze pro své potřeby. V následující kapitole se budeme věnovat její instalaci, popisu funkcí a využití, které vychází především z praktických poznatků.

4.4. Instalace a využití A2billing

Instalaci je možné provést na jakékoli distribuci Linuxu. A2billing pro svůj chod potřebuje webový server s podporou php5 a databázi. Konkrétní požadavky jsou:

- webový server Apache2
- php5
- MySQL 5 nebo PostgreSQL 8

Předpokladem je správně fungující webový server s podporou php5, zprovozněná základní konfigurace Asterisku a dostupný databázový server. V našem konkrétním případě využijeme již dříve zprovozněnou MySQL databázi. Popisem instalace a konfigurací webového serveru se nebudeme v této práci zabývat. Podrobně popsany popis instalace webového serveru najdeme v literatuře [2].

Otestujeme webový server tím, že napíšeme IP adresu nebo doménové jméno do prohlížeče. Můžeme také vytvořit jednoduchý PHP skript k základnímu otestování. Pokud jsme postupovali správně a webový server odpovídá korektně na dotazy, můžeme přejít k instalaci A2Billing.

4.4.1. Stažení zdrojových souborů

Musíme stáhnout zdrojové soubory ze stránek <http://www.asterisk2billing.org/> Zdrojové soubory můžeme stáhnout přes webový prohlížeč nebo pokud máme nainstalován svn klienta, tak můžeme stáhnout zdrojové soubory přímo z repozitářů. Pro stažení zdrojových souborů z repozitářů je nutné se zaregistrovat. Zdrojové soubory byly staženy přes webový prohlížeč a nahrány na server pomocí scp do adresáře /usr/local/src/a2billing. Stažený archiv je nutné rozbalit.

4.4.2. Příprava databáze

Aplikace A2billing má k dispozici předpřipravené SQL skripty pro vytváření uživatelského účtu. Prvním skriptem, který spustíme z příkazové řádky je skript a2billing-createdb-user.sql, který je umístěn v adresáři /usr/local/src/a2billing/DataBase/Mysql-5.x.

Jeho použití v příkazovém řádku je následující.

```
# mysql -u root -p"root password" < a2billing-MYSQL-createdb-user.sql
```

Jelikož uživatel root má přístup do konzole Mysql databáze bez hesla můžeme vynechat parametry -u, -p. Příkaz se tedy zjednoduší na pouhý import dat z SQL souboru.

```
# mysql < a2billing-MYSQL-createdb-user.sql
```

Skript tedy vytvořil uživatele a2billinguser s heslem a2billing a dále pak prázdnou databázi mya2billing. Druhým skriptem, který je nutné spustit je skript instal-db.sh, který je umístěn ve stejném adresáři jako předchozí skript. Skript provede vytvoření tabulek (97) a import dat. Spustíme jej přímo z příkazového řádku.

```
# /usr/local/src/a2billing/DataBase/mysql-5.x/install-db.sh
```

Skript se zeptá na parametry připojení jako je IP adresa databázového serveru, jméno uživatele, jméno databáze a heslo a následně vytvoří tabulky a importuje data. Po dokončení skriptu je vhodné ověřit, jestli se vytvořilo všech 97 tabulek.

```
#mysql mya2billing  
mysql> show tables;
```

4.4.3. Vytvoření konfiguračního souboru A2billing

Aplikace A2billing má pouze jeden konfigurační soubor, který je nutné zkopírovat do adresáře /etc. Soubor najdeme v adresáři /usr/local/src/a2billing/a2billing.conf

Jediné nastavení, které musíme změnit je nastavení přístupu k databázi a to vypadá následovně.

```
[database]  
hostname = localhost  
port = 3306  
user = a2billinguser  
password = a2billing  
dbname = mya2billing  
dbtype = mysql
```


4.4.4. Úprava práv a rozšíření konfigurace Asterisku

Aplikace A2billing se snaží oddělit konfigurační soubory od systému Asterisk tak, aby konfigurace byla přehledná. SIP a IAX uživatelské účty v aplikaci A2billing se neukládají přímo do konfiguračních souborů sip.conf a iax.conf. Tyto konfigurační soubory se rozšiřují o další soubory, které jsou pak využívány aplikací A2billing. Dále je pak nutné upravit práva tak, aby A2billing mohl zapisovat do příslušných konfiguračních souborů.

```
chmod 777 /etc/asterisk
touch /etc/asterisk/additional_a2billing_iax.conf
touch /etc/asterisk/additional_a2billing_sip.conf
echo \#include additional_a2billing_sip.conf >> /etc/asterisk/sip.conf
echo \#include additional_a2billing_iax.conf >> /etc/asterisk/iax.conf
chown -Rf www-data /etc/asterisk/additional_a2billing_iax.conf
chown -Rf www-data /etc/asterisk/additional_a2billing_sip.conf
```

4.4.5. Konfigurace Asterisk Manager

Aplikace a2billing potřebuje mít přístup do konzole Asterisku, aby mohla zjistit aktuální počet přihlášených uživatelů a dále pak mohla monitorovat probíhající hovory. Konfigurační soubor /etc/asterisk/manager.conf změníme následovně.

```
[general]
enabled = yes
port = 5038
bindaddr = 0.0.0.0

[myasterisk]
secret=mycode
read=system,call,log,verbose,command,agent,user
write=system,call,log,verbose,command,agent,user
```

4.4.6. Instalace AGI komponent

AGI komponenty jsou stěžejní částí aplikace A2billing. Popisu komponenty se budeme věnovat v následujících kapitolách. Před instalací těchto komponent je třeba znát verzi Asterisku. Pro starší verze Asterisku (1.2 a 1.4) jsou určeny rozdílné konfigurační soubory, které nejsou vzájemně kompatibilní s novější verzí Asterisku (1.6). V této práci využívám Asterisk verze 1.6. Instalaci provedeme do adresáře /usr/share/asterisk/agi-bin.

Do tohoto adresáře je potřeba přepokopírovat několik souborů ze zdrojového adresáře /usr/local/src/a2billing/AGI.

```
cp /usr/local/src/a2billing/AGI/a2billing.php /usr/share/asterisk/agi-bin/
```

Dalším krokem je rozšíření volacího plánu o soubor `extensions_a2billing.conf`. Do souboru `/etc/asterisk/extensions.conf` přidáme jeden z následujících řádků. Podle toho, kterou verzi Asterisku využíváme.

```
#include extensions_a2billing.conf
#include extensions_a2billing_1_6.conf
```

Zkopírujeme konfigurační soubory do adresáře `/etc/asterisk`. Pokud využíváme Asterisk verze 1.6 je nutností zkontrolovat a popřípadě nahradit všechny znaky `"|"` znakem `","`. Bez této úpravy nebude aplikace `a2billing` pracovat korektně.

4.4.7. Konfigurace pravidelných úloh

Aplikace pravidelně odesílá emaily a informuje tak zákazníky například v případě nízkého kreditu pro volání nebo také generuje faktury. V aplikaci je několik úloh, které se provádí pravidelně. Upravíme soubor `/etc/crontab`, který se v operačním systému Linux stará o pravidelné úlohy. Do tohoto souboru přidáme následující řádky.

```
#upgrade tabulky s přepočtem měnových kurzů - denně
0 6 * * * php /usr/local/src/a2billing/Cronjobs/currencies_update_yahoo.php
```

```
#řízení služeb měsíčního předplatného
0 6 1 * * php /usr/local/src/a2billing/Cronjobs/a2billing_subscription_fee.php
```

```
#kontrola každého uživatele každou hodinu a odeslání emailu v případě, že zůstatek kreditu je
menší než potřebný
0 * * * * php /usr/local/src/a2billing/Cronjobs/a2billing_notify_account.php
```

```
#skript projde všechny předvolby, které má zákazník rezervovány a zkontroluje zda má za tyto
předvolby zákazník něco zaplatit.
0 2 * * * php /usr/local/src/a2billing/Cronjobs/a2billing_bill_diduse.php
```

```
#tento skript se stará o opakující se služby
0 12 * * * php /usr/local/src/a2billing/Cronjobs/a2billing_batch_process.php
```

```
#generování faktur
0 6 * * * php /usr/local/src/a2billing/Cronjobs/a2billing_batch_billing.php
```

```
#řízení automatického vytáčení
```

```
* / 5 * * * * php /usr/local/src/a2billing/Cronjobs/a2billing_batch_autodialer.php
```

```
# řízení alarmů
```

```
0 * * * * php /usr/local/src/a2billing/Cronjobs/a2billing_alarm.php
```

4.4.8. Instalace webových aplikací

Další částí aplikace je webové rozhraní, které slouží ke konfiguraci a správě uživatelských účtů, výběru tarifkace a dalším administrátorským účelům. Instalaci provedeme do domovského adresáře webového serveru, do kterého zkopírujeme aplikace a upravíme práva.

```
#mkdir /var/www/a2billing
```

```
#cp -rf /usr/local/src/a2billing/admin /var/www/a2billing
#cp -rf /usr/local/src/a2billing/agent /var/www/a2billing
#cp -rf /usr/local/src/a2billing/customer /var/www/a2billing
#cp -rf /usr/local/src/a2billing/common /var/www/a2billing
#chown www-data:www-data /var/www/a2billing
```

Nyní se přihlásíme do administrátorské aplikace. Otevřeme internetový prohlížeč a zadáme adresu serveru s příslušným kontextem. V našem případě `http://192.168.10.5/a2billing/admin`

K přihlášení využijeme uživatele `root` a standardní heslo je nastaveno na „`changepassword`“. Z bezpečnostního hlediska po přihlášení ihned změním heslo. Pokud jsme postupovali podle návodu, tak po přihlášení by měla úvodní obrazovka vypadat následovně.

4.5. Konfigurace A2billing

4.5.1. Uživatelské kategorie

Aplikace integruje ve své hierarchii celkem tři uživatelské kategorie. První kategorie jsou administrátoři, kteří mají přístup ke konfiguracím a CDR záznamům všech uživatelů. Tyto konfigurace mohou upravovat a vytvářet další. Jsou to uživatelé s největším oprávněním.

Další kategorie jsou agenti. Agenti si můžeme představit jako vedoucího sekce nebo oddělení. Tento uživatel má možnost nahlížet na CDR záznamy svých podřízených. Má tedy přehled o výši účtu svých podřízených.

Poslední kategorií jsou uživatelé. Ti mají právo nahlížet na svoje CDR záznamy a upravovat základní informace o své osobě. Dále mají možnost zakoupení kreditu přes služby Paypal a jiné.

4.5.2. Trunky neboli směrování hovoru

Pro zprovoznění jakéhokoliv volání je nutné mít nastaveny trunky. Nastavují se v sekci „Providers“. Nejprve je důležité definovat jméno a popis poskytovatele služeb. K tomuto poskytovateli se přidá „trunk“ neboli cesta. V konfiguraci cest je nutné zvolit technologii, na které poskytovatel služeb pracuje (SIP, IAX2, ZAP, H323). Dalším parametrem je IP adresa poskytovatele. Pro vnitřní volání pouze na úrovni Asterisku vyplníme %dialingnumber%. Pokud přidáváme veřejného poskytovatele služeb, tak zadáme jeho doménové jméno VoIP brány. Stejná konfigurace se musí objevit i v konfiguračním souboru /etc/asterisk/sip.conf, kde přesněji definujeme poskytovatele služeb (uživatelské jméno, heslo a metoda autentizace). Musíme si dát pozor na názvy – musí být shodné. Detailní konfigurace a možnosti jsou zobrazeny v tabulce. Dobrou možností aplikace A2billing je přesměrování hovorů na jiného poskytovatele v případě problémů s primárním poskytovatelem.

voip-provider	jméno poskytovatele služeb
label	unikátní jméno
add prefix	
remove prefix	
provider techn	SIP, IAX2 , ZAP, H323
provider IP	%dialingnumber%
additional parameter	
failover trunk	
corrent connections	0
Maximum conn	-1
ifmaximum conn	
status	Active

4.5.3. Sazby za volání

Abychom mohli využít trunky, musíme nakonfigurovat tarify. Tarify nastavujeme v sekci „rates“. Před vytvořením cen za volání, vytvoříme takzvané „rate cards“. A2billing dokáže rozpoznat prefix pro mezinárodní předvolby a také prefix pro lokální operátory (mobilní a pevné telefonní linky). A2billing přiřadí hovoru konkrétní tarif právě podle prefixů. Nastavení tarifů je velmi detailní. Lze nastavit poplatek za spojení a poplatek za ukončení hovoru. Máme také možnost zvolit vyúčtování po sekundách nebo za konkrétní časový interval. Další možností je účtovat například první minutu celou a pak po 30 vteřinách. Možností konfigurace se aplikace vyrovná reálnému nasazení. Dokáže definovat různé tarify pro různé časy. Další možností je zvolit jinou cenu nákupu a jinou prodejní cenu. Aplikace z těchto hodnot dokáže spočítat čistý zisk za hovor. V tomto je A2billing opravdu výborný software, který odráží reálné prostředí. Samozřejmostí v této sekci je import cen z CSV souboru.

4.5.4. Vytvoření zákazníků

Abychom mohli vytvořit jakýkoli SIP nebo IAX účet je nejprve nutné definovat zákazníka. Množina zákazníků má dvě podmnožiny a těmi jsou podmnožina SIP a IAX. Jeden zákazník tedy může využívat dva protokoly podle typu svého klientského zařízení. Hlavní identifikátorem zákazníka je číslo účtu (account number). Toto číslo je automaticky vygenerováno aplikací. Také jsou vygenerovány zákaznickovy přihlašovací údaje pro webové rozhraní. Další položky jsou intuitivní. Administrátor má možnost i zablokovat uživatele, není potřeba jej hned vymazat.

4.5.5. Vytvoření SIP zákazníka

Pokud vytvoříme nového zákazníka, můžeme automaticky vytvořit SIP konfiguraci, která se ukládá do souboru `/etc/asterisk/additional_a2billing_sip.conf`. U každého zákazníka je možné definovat metodu platby. Metodou platby je myšleno:

- prepaid – platba pomocí předplaceného kreditu
- postpaid – platba fakturou za určité časové období

A2billing v tomto případě věrně kopíruje realitu nákupu nového telefonního čísla. Když si zákazník koupí v obchodě nové telefonní číslo s předplaceným kreditem, většinou dostane také základní kredit pro volání. V aplikaci je tedy možné definovat počáteční kredit na volání. Pokud si zákazník koupí paušální tarif, tak platí paušál, ve kterém může mít definován určitý počet volných minut. V aplikaci je to řešeno pomocí balíčků, ve kterých definujeme konkrétní počet volných minut a přiřadíme je konkrétnímu zákazníkovi.

A2billing nevyužívá konfiguraci z databáze, ale textové konfigurace SIP účtů ve výše zmiňovaném souboru `/etc/asterisk/additional_a2billing_sip.conf`. Telefonní číslo definujeme tedy až na úrovni konfigurace Asterisku.

```
[4440]
username=4634967440
name=4440
accountcode=4634967440
type=friend
secret=heslo
host=dynamic
context=a2billing
allow=alaw
trustpid=yes
sendrpid=yes
canreinvite=no
```

qualify=yes

V předešlé konfiguraci je tedy zákazník s ID 4634967440, telefonní číslo má však přiřazeno 4440. Pomocí atributu account code aplikace rozpozná telefonní číslo zákazníka. V aplikaci potom naúčtuje zákazníkovi cenu za hovor podle příslušného tarifu.

4.5.6. CDR záznamy v A2billing

Aplikace poskytuje poměrně rozsáhlé statistiky a grafy, které jsou vytvořeny na základě CDR záznamů. Statistiky a ceny je možné zobrazit pouze na konkrétního zákazníka nebo také na skupinu zákazníků neboli organizační jednotku. Dále je možnost si zobrazit vytížení jednotlivých trunků a to v požadovaných intervalech. Další možností je porovnávání počtu hovorů za určitá časová období. Statistiky aplikace jsou na velmi vysoké úrovni. Pokud by korporace měnila poskytovatele služeb, tak tyto statistiky by hrály hlavní roli při výběru nového poskytovatele, protože ze statistik lze zjistit v kterých dnech a časech je ústředna nejvytíženější. Výpis hovorů vypadá následovně.

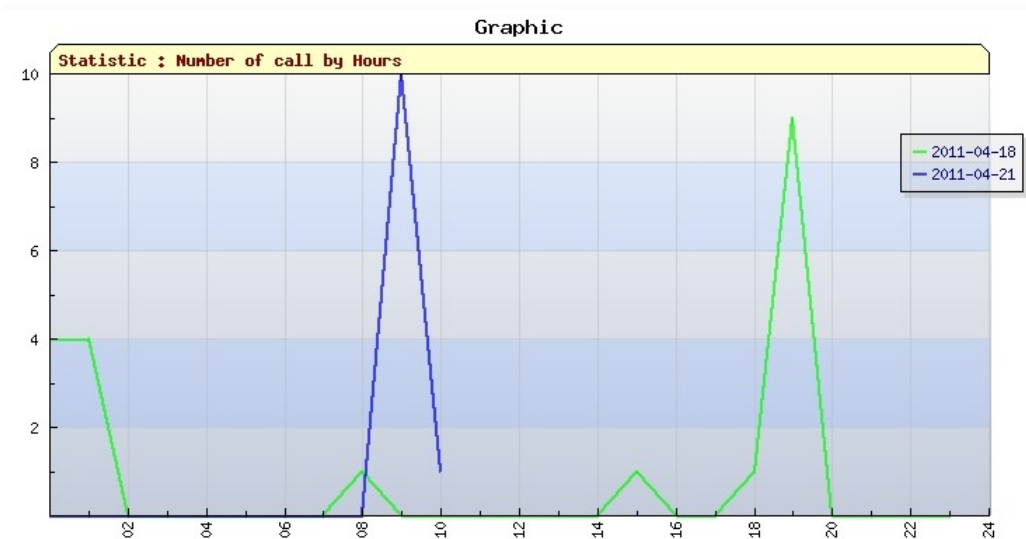
- Call Logs -																
	Date	CallerID	DNID	Phone Number	Destination	Buy Rate	Sell Rate	Duration	Account	Trunk	TC	CallType	Buy	Sell	Margin	Markup
1.	2011-05-03 09:43:31	4441	4440	4440	0	0.010 USD	0.020 USD	00:41	6937054692	local	ANSWER	STANDARD	1.025 USD	2.050 USD	50.00%	100.00%
2.	2011-05-03 09:42:13	4441	4440	4440	0	0.010 USD	0.020 USD	00:44	6937054692	local	ANSWER	STANDARD	1.100 USD	2.200 USD	50.00%	100.00%
3.	2011-05-03 09:40:07	4440	4441	4441	0	0.010 USD	0.020 USD	01:22	4634967440	local	ANSWER	STANDARD	2.050 USD	4.100 USD	50.00%	100.00%
4.	2011-05-03 09:35:49	4441	4440	4440	0	0.010 USD	0.020 USD	00:07	6937054692	local	ANSWER	STANDARD	0.012 USD	0.014 USD	16.64%	19.97%
5.	2011-05-03 09:35:10	4441	4440	4440	0	0.010 USD	0.020 USD	00:23	6937054692	local	ANSWER	STANDARD	0.038 USD	0.046 USD	16.67%	20.01%
6.	2011-05-03 09:28:31	4440	4441	4441	0	0.010 USD	0.020 USD	00:05	4634967440	local	ANSWER	STANDARD	0.008 USD	0.010 USD	16.70%	20.05%

Z výpisu je možné zjistit číslo volajícího, na které telefonní číslo volal, jak dlouho hovor trval a kolik hovor stál, popřípadě jaký měl poskytovatel služeb z hovoru zisk.

4.5.7. Porovnání volání

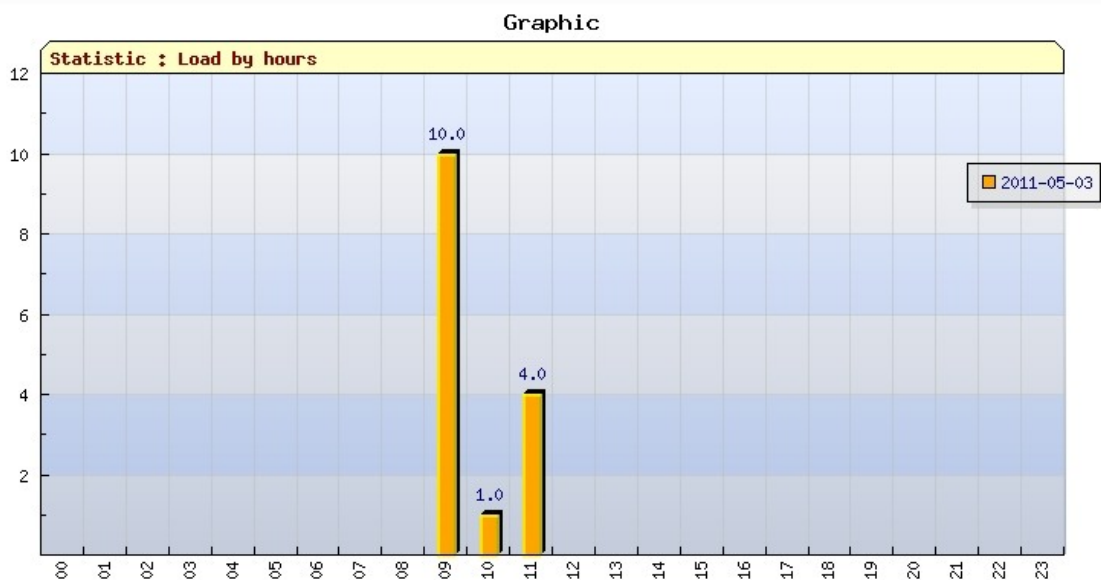
Záložka porovnat volání dokáže vytvořit grafy o volání závislé na počtu provolaných minut za hodinu, počtu volání za hodinu a cenovém zisku za hodinu. Porovnávat volání lze maximálně ze 4 dnů v libovolném datu. V horní části obrazovky je zobrazený podobný filtr jako na předešlých záložkách. Pod filtrem je zobrazena tabulka, která porovnává data s počty volání, průměrnou dobou spojení, celkovým prodejem, celkovým nákupem a celkovým ziskem.

Nevýhodou této statistiky je porovnání maximálně za poslední čtyři dny. Nemáme tedy možnost srovnávat různé dny v roce a různé týdny.



4.5.8. Denní a měsíční statistiky

Tato položka obsahuje filtr. Je možno vybrat konkrétní den, zákazníka, trunk nebo číslo volaného. Z této statistiky lze zjistit kritické hodiny, ve kterých je ústředna nejvytíženější. Tyto statistiky jsou vhodné především pro administrátora. Ten má tak přehled, ve kterých hodinách je možné dělat údržbu serveru, popřípadě plánovat dočasné odstávky.



Poslední možné statistiky jsou měsíční. V této statistice lze sledovat přírůstky nebo naopak úbytky volání za jednotlivé měsíce s historií až 6 měsíců zpětně.

4.5.9. Jazykové mutace A2billing

Aplikace a2billing je přeložena do několika světových jazyků. Základem je angličtina následují další světové jazyky španělština, francouzština, portugalština. Program byl v minulosti překládán panem Ing. Danielem Jandou a čeština byla součástí starších verzí. Aktuálně jsou k dispozici pouze světové jazyky.

4.5.10. Shrnutí A2billing

A2billing je rozsáhlý projekt. Aplikace má především svoje přednosti v tarifkaci. Možnosti nastavení tarifů a účtování za hovory jsou opravdu na velmi vysoké úrovni. Také statistiky jsou přehledné. Aplikace je spíše určena pro poskytovatele VoIP služeb, kterým může velmi pomoci s konfigurací tarifů a se správou uživatelů. Problémem tohoto projektu je dokumentace. Návod k instalaci projektu je velmi zastaralý a nepřesný. Vlastní dokumentace se zabývá pouze několika málo kroky. Vývojový tým předpokládá, že ten, kdo bude chtít aplikaci využívat, si koupí placenou podporu a nechá si aplikaci nakonfigurovat podle vlastních požadavků. Dokumentace je tedy nedostatečná. Jediným dobrým zdrojem informací o aplikaci a řešení problémů je fórum na adrese <http://forum.asterisk2billing.org/>. V tomto fóru jsem našel odpovědi na většinu problémů při instalaci a konfiguraci aplikace. Druhou polovinu problémů jsem vyřešil procházením log souborů a hledáním řešení přes internetové vyhledávače.

Asterisk umožňuje mít veškerou konfiguraci SIP účtů uloženou v databázi. A2billing využívá stále textově orientované konfigurační soubory. Neukládá tedy konfigurační soubory do databáze, ale zapisuje konfigurace do souboru. Při větším počtu uživatelů se soubor s uživateli stává nepřehledný. Velkou výhodou by tedy bylo mít konfigurace SIP uživatelů v databázi.

Největším problémem při implementaci projektu byly AGI knihovny. Oficiálně je v dokumentaci uvedeno, že A2billing využívá konfiguraci z textového souboru uloženého na pevném disku. Pravdou ovšem je, že A2billing si konfiguraci AGI vytahuje z databáze.

5. Notifikace při překročení limitů nákladů

V posledních letech je celosvětovým problémem zneužívání VoIP telefonních ústředn. Softwarové telefonní ústředny jsou při takovém útoku zneužity pro volání na drahá telefonní čísla a způsobují tak majitelům velké finanční škody. Během útoku se útočník napojí na telefonní ústřednu a provede několik desítek popřípadě stovek hovorů na drahá telefonní čísla v celém světě. Tímto způsobem se může majiteli telefonní ústředny navýšit platba až o statisíce korun. Nastává tedy otázka jak útokům předejít, popřípadě minimalizovat ztráty, pokud by došlo ke zneužití. V následující kapitole řeším možnosti prevence tohoto problému pomocí zabezpečení Asterisku. Druhou alternativou je automatická kontrola CDR záznamů v databázi a včasná notifikace administrátora v případě zneužití VoIP služeb. V této kapitole jsou shrnuty poznatky s vlastní implementací tohoto řešení.

5.1. Zabezpečení SIP a RTP protokolu

Základem obrany proti útokům na VoIP ústřednu je zabezpečení telefonní ústředny a pravidelná aktualizace programových balíčků. SIP standardně funguje přes UDP protokol na portu 5060, ale může být taktéž nakonfigurován pro komunikaci na TCP. Asterisk od verze 1.6 podporuje zprovoznění signálního protokolu SIP přes TCP. Tímto se otevírá možnost využít zabezpečení protokolu SIP přes TLS (Transport Layer Security). Asterisk nabízí tuto možnost přibližně od roku 2008. Při využití TLS je nutné vygenerovat certifikáty pro server i pro klienty. Klientské certifikáty je třeba bezpečným způsobem dopravit na stranu klienta. Mnoho ústředn však zatím tuto možnost zabezpečení nevyužívá, protože v Asterisku je poměrně nová a dále tato metoda zajišťuje pouze zabezpečení signálního protokolu. Samostatný hlas přenášený přes RTP (real-time transport protocol) pakety je tedy nešifrován.

Na řadu tedy přichází SRTP (Secure real-time transport protocol). Podpora SRTP je v Asterisku od verze 1.8, která byla vydána na podzim roku 2010. V posledních letech se zabezpečení signálního a transportního protokolu posunulo směrem dopředu, stále však mnoho klientských zařízení tyto novější standardy nepodporují a tak mnoho korporací řeší bezpečnost vlastním způsobem. Před zneužitím VoIP ústředny se chrání firewallem a využitím VoIP technologie pouze ve vnitřní síti. Pokud je nutné připojit klienty, kteří jsou mimo vnitřní síť, tak nastává možnost využití IPSec nebo jiné VPN technologie. Ochrana proti zneužití VoIP telefonní ústředny nemusí být postavena pouze na šifrování. Jsou případy, kdy šifrování použít nelze a to především u poskytovatelů VoIP služeb. V tomto případě je možné využít právě účtovacího systému, který včas upozorní na zneužití VoIP ústředny, nebo také upozorní v případě, že daný uživatel překročil limity pro volání.

5.2. Pravidelná kontrola CDR záznamů

V této kapitole předpokládám, že Asterisk ukládá CDR záznamy do databáze MySQL ve standardním formátu a také jsou ukládány doplňkové atributy, jako jsou IP adresy. Tato data

lze v pravidelném intervalu kontrolovat. Nabízí se tedy možnost vytvoření vlastního skriptu, který v pravidelných intervalech bude tuto databázi dotazovat, vypočte celkovou dobu hovoru u každého klienta a v případě nadlimitního volání předá notifikaci administrátorovi serveru. Další kroky jsou v tomto případě na administrátorovi a podpoře ze strany VoIP poskytovatele. Tímto způsobem lze řešit situaci v případě zneužití VoIP služeb. Skript této situaci nedokáže nepředejít, pouze upozorní, že k této situaci došlo a včas informuje.

5.2.1. Bash

Bourne Again SHell je standardní interpret příkazů v Linuxu založený na Bourne shell. Funguje jako rozhraní mezi uživatelem a operačním systémem. Jelikož je součástí GNU projektu, nebylo problémem ho importovat na unixové systémy, takže se jeho znalost uplatní i mimo Linux. Je to také velice mocný programovací nástroj. Když nám chybí nějaký program, nemusíme ho hned psát v kompilovaném jazyku (C, C++, Java), ale je možné vyřešit náš problém vytvořením skriptu.

5.2.2. Cron

Cron je softwarový démon, který v operačních systémech automatizovaně spouští v určitý čas nějaký příkaz resp. proces (skript, program apod.). Jedná se o specializovaný systémový proces, který v operačním systému slouží jakožto plánovač úloh, jenž umožňuje opakované spouštění periodicky se opakujících procesů. Cron využijeme tedy jako spouštěč vlastního skriptu.

Zápis v cronu může vypadat následovně.

```
0 12 1 * * root /skript.sh
```

První znak zápisu značí minuty (0-59), druhý znak hodiny (0-23), třetí znak den v měsíci (1-31), čtvrtý znak měsíc (1-12) poslední znak den v týdnu (0-6). Týden v tomto případě začíná nedělí (0). Dalším atributem je uživatel, který bude skript spouštět a poslední je cesta ke skriptu. V našem příkladě tedy budeme spouštět skript vždy první den v měsíci ve 12 hodin.

5.2.3. Návrh skriptu

Asterisk ukládá CDR záznamy do tabulky cdr a to hned po každém ukončeném hovoru. V databázi tedy máme přehled nad tím kdo, kdy a kam volal. Základními parametry pro výpočet jsou tedy „billsec“ a identifikace „clid“. Jak již bylo řečeno je to identifikace volajícího, druhým podstatným parametrem je „billsec“. Tento parametr určuje, jak dlouho hovor probíhal a třetím aspektem je datum a čas hovoru, který je reprezentován v databázi položkou „calldate“. Skript

tedy bude mít za úkol u každého účastníka spočítat celkovou dobu hovoru za určité časové období a tento čas porovnat s aktuálně nastaveným limitem. V případě překročení limitu zaslat upozornění.

Jediným problémem tohoto řešení je, že Asterisk zapisuje CDR záznamy až po skončení hovoru. V případě, že skriptem chceme kontrolovat databázi každou hodinu, tak nemusí námi navrhované řešení fungovat.

Z bezpečnostního hlediska tedy musíme omezit maximální čas hovoru a po určitém čase hovor ukončit. Toto je jediná možnost jak lze implementovat zabezpečení z pohledu CDR záznamů.

V číslovacím plánu tedy definujeme maximální dobu hovoru. Maximální dobu spojení tedy stanovíme na 6600 sekund (1hodina 50minut). Ten může vypadat potom následovně.

```
[sip]
exten => _X.,1,Answer
exten => _X.,n,Set(TIMEOUT(absolute)=6600)
exten => _X.,n,Dial(SIP/${EXTEN})
exten => _X.,n,Hangup()
```

Hovor tedy v našem případě skončí po uplynutí časového intervalu a Asterisk zapíše do databáze záznam o hovoru. Na základě těchto dat potom skript funguje korektně.

5.2.4. Entita result

Úloha této entity je především pro dočasné výpočty. Entita má pouze dva atributy a tím jsou:

Atribut	Datový typ
clid	varchar(60)
sumsec	int(11)

- clid – identifikace volajícího
- sumsec – celkový účtovaný čas za poslední hodinu

5.2.5. Realizace skriptu

Skript jsem realizoval v Bash. Skript se připojí do databáze MySQL a SQL dotazem vypočte počet provolaných sekund za poslední hodinu. Tato data uloží do tabulky „result“. Skript je naplánován v cronu na každé dvě hodiny.

```
INSERT result (SELECT clid,SUM(billsec)
FROM asterisk.cdr
```

```
WHERE calldate > DATE_ADD(NOW(), INTERVAL - 2 HOUR)
GROUP BY clid);
```

Výsledná data v tabulce „result“ mohou vypadat následovně.

Entita result

clid	sumsec
3459	123645
3412	521230

Nad těmito daty se následně provádí porovnání s aktuálně nastaveným limitem. Limit jsem stanovil hodinu a 45minut, ale tento limit je velmi variabilní a je možné jej kdykoli upravit. Pokud by šlo například o realizaci Asterisku pro telefonní call centrum, volil bych spíše parametry vyšší. Nebo úplně naopak tento skript v pracovní době vyřadit a nechat jej spuštěný pouze mimo pracovní dobu. Ověření limitů se provede následujícím SQL dotazem.

```
SELECT clid
FROM asterisk.result
WHERE sumsec > 6300;
```

Na výstupu tedy máme identifikace volajícího. Výstup toho SQL dotazu přesměrujeme do textového souboru, který následně přepošleme emailovou zprávou na adresu administrátora. Email se posílá pouze tehdy, když opravdu jsou překročené limity. Respektive se kontroluje velikost souboru. V případě nenulové velikosti se email odešle.

```
file="./result.txt"
```

```
if [ -s $file ]; then
echo "Problem with many calls at the same time. Callerid is in attachement" | mutt -s "Asterisk
warning" admin@email.cz -a result.txt
else
exit
fi
```

Administrátor serveru tedy dostane upozornění, ve kterém jsou zobrazena čísla, která volají nad rámec limitů. Závěrečným krokem je výmaz dat z tabulky „result“.

Tento skript nemusí fungovat pouze jako upozornění emailem na případné zneužití VoIP služeb. Po úpravě časového intervalu lze sledovat provolané minuty u zákazníka a takovýmto jednoduchým způsobem ho kontaktovat v případě, že provolal stanovený limit.

Skript dále může být rozšířen o zastavení služeb VoIP ústředny v případě velmi vysokého počtu hovorů. Způsobů jak spočítat kolik účastníků překračuje limit je několik.

Jedním ze způsobů je využití programů v Linuxu a nepřetěžování databázového serveru. Program „wc“ spočítá řádky souboru a následně tuto hodnotu porovná s limitem.

```
# cat ./result.txt | wc -l
```

Vrátí počet řádků souboru, do kterého jsme uložili nadlimitní identifikace uživatelů. Výsledkem je tedy pouze číslo. Pro stanovení limitu musíme znát celkový počet uživatelů ústředny. Pokud je počet volajících nad limit, skript může jednoduše zastavit služby VoIP ústředny a informovat administrátora serveru o zastavení služby. Takovéto řešení přichází v úvahu pouze tehdy, pokud server nemá 24 hodinový dohled. Útočníci většinou využívají nočních hodin ke zneužití VoIP ústředny. O zastavení služeb bych uvažoval opravdu jen tehdy, kdyby počet zneužitých účtů přesahoval více než 80%.

V některých případech je absolutně nepřijatelné zastavení VoIP služeb. V Asterisku můžeme vynutit odhlášení uživatele přes příkazový řádek a to tímto způsobem.

```
# asterisk -rx "sip unregistred clid"
```

Toto řešení je ovšem dočasné a pouze uživatele na krátkou dobu odhlásí. Uživatel má možnost se znovu přihlásit. Toto řešení neblokuje uživatele.

Asterisk v současné době nemá možnost zablokovat uživatele přes příkazový řádek. Existuje ovšem jiný způsob jak blokovat uživatele.

Předpokládáme, že VoIP ústředna využívá pro ochranu firewall založený na iptables. V databázi u každého hovoru máme uloženy IP adresy a to jak veřejnou tak interní IP adresu. Můžeme tedy rozšířit skript o blokování uživatelů na základě IP adres. SQL dotazem zjistíme IP adresy uživatelů, kteří mají nadlimitní délky hovorů a tyto IP adresy předáme jako parametry pro iptables.

Pokud nějakým způsobem implementujeme monitoring hovorů, tak jen můžeme snížit následky. Vždy však platí, že většině problémů se dá předcházet, pokud si uvědomujeme riziko a cenu zneužití VoIP služeb. Základem tedy budou silná bezpečná hesla a vhodný způsob zabezpečení.

Takto řešené skripty se opírají o záznamy v databázi. Ty jsou do databáze vždy uloženy po skončení hovoru. V podmínkách vytvoření skriptu bylo nutné nastavit maximální dobu hovoru. Po uplynutí tohoto časového limitu je hovor ukončen a CDR záznam zapsán do databáze.

Toto řešení není úplně ideální, protože omezuje zákazníka na určitý časový limit. Po vypršení tohoto limitu musí zákazník vytočit číslo znovu. Pokud bychom nechtěli nastavovat maximální dobu hovoru, tak existuje druhá varianta jak zjistit kdo a jak dlouho aktuálně volá. Možnost je využít příkazové řádky Asterisku a to příkazem „sip show channelstats“. Výstup tohoto příkazu může vypadat následovně.

```
Peer      Call ID    Duration Recv: Pack Lost    ( %) Jitter Send: Pack Lost    ( %) Jitter
192.168.10.22  0571bbcb02f 00:01:44 00000000004 00000000000 ( 0.00%) 013637 00000000001 0015605604 (1560560400.00%)
192.168.10.12  YzNmMWEyNzF 00:01:44 00000000138 00000000000 ( 0.00%) 014405 0000000139 0005736405 (4126910.07%)
2 active SIP channels
```

Výstupem tohoto příkazu jsou aktuálně probíhající hovory a právě v tomto případě je zobrazena aktuální délka hovoru. Výstup tohoto skriptu je možné uložit do textového souboru. S textovým souborem potom můžeme pracovat pomocí programů, jako jsou AWK nebo SED.

5.3. Kontrola počtu spojení za minutu

Jak jsem již uvedl v předešlé kapitole, tak nejčastějším zneužití VoIP služeb jsou dlouhé hovory na zahraniční čísla. Další možností jak navýšit účet za volání je poplatek za spojení. Ten je ve většině případů účtován ihned po vytvoření spojení. Tento poplatek je na některá zahraniční čísla a především na satelitní telefony velmi vysoký. Ústředna v takovém případě může vytvářet několik desítek spojení během několika málo minut. V tomto případě nemůže atribut „billsec“ přesáhnout stanovený limit a přesto účet za služby hovorného poroste velkým tempem.

Princip řešení je velmi podobný předešlému příkladu. Stejným způsobem se dotážeme databázového serveru, ale místo celkového účtovaného času. Spočítáme počet záznamů u každého uživatele za konkrétní časový limit. Do cronu tedy přidáme další úlohu, kterou budeme provádět každých 30 minut. Teoretickým výpočtem bylo zjištěno, že útočník může generovat každých 5 sekund jeden krátký telefonát přes jeden uživatelský účet. Za 30min je to přibližně 360 pokusů o spojení. Nebudeme brát zřetel na to, jestli hovor byl zodpovězen nebo nebyl. V případě, že je počet pokusů o spojení více než například 60 na jednom uživatelském účtu, je toto vyhodnoceno jako podezřelé chování a skript zašle notifikaci administrátorovi s informací, který z uživatelů generuje nadměrné množství hovorů za krátký časový úsek. Limit se bude lišit na základě povahy využití pobočkové ústředny. Jiné limity bude mít nastaveno call centrum, kde hovorů během krátkého časového intervalu může být více než uváděných 60.

5.4. Další možnosti notifikací

Základem administrace každého serveru je pravidelná kontrola logů a pravidelná údržba serverů. Mezi tuto pravidelnou údržbu patří především instalace aktualizací a všech bezpečnostních oprav. V Asterisku se v pravidelných intervalech objevují záplaty, které

opravují bezpečnostní chyby. Administrátoři VoIP technologií by však neměli tuto pravidelnou údržbu opomíjet. Z hlediska časového vytížení však nezbyvá na kontrolu logů dostatečný čas.

Částečným řešením může být dohledový systém, nebo jednoduchá aplikaci na monitoring VoIP infrastruktury. Nagios je v posledních letech nejpoužívanější monitorovací systém. Tím, že nagios je open source projekt, může být rozšířen o kontrolu Asterisku, popřípadě SIP modulu v Asterisku. Možností monitoringů je hned tedy několik.

Pokud však Asterisk používáme v ostrém provozu, je dobré implementovat více zabezpečovacích a notifikačních metod. Neměli bychom se pouze zaměřit na jednu technologii a spoléhat pouze na ni.

5.5. Notifikace v A2billing

Jak jsem již zmínil, tak jsou dva možné způsoby tarifkace za služby hovorného. První možnost je prepaid, neboli takzvaná platba pomocí předplacených karet. V případě, že máte nedostatečný kredit pro volání, tak a2billing hovor nespojí. Telefonní automat vám odpoví, že máte nedostatečný kredit a hovor ukončí.

Druhá možnost je postpaid, kdy zákazník platí měsíční faktury, které aplikace dokáže vygenerovat automaticky. A2billing odráží současnou realitu a je tedy možné si nastavit limity.

A2billing taktéž dokáže upozornit zákazníka emailem nebo během hovoru, že jeho kredit na volání se snížil na určitou hodnotu.

6. Závěr

Cílem práce bylo nelézt vhodné produkty pro účtování v prostředí Asterisk PBX. Nejprve bylo nutné nastudovat danou problematiku. Ta zahrnovala vlastní zprovoznění systému Asterisk pod operačním systémem Linux. Jako operační systém byl zvolen Debian 6. Na tento operační systém byl nainstalován Asterisk. Verze Asterisku zvolená při začátku této práce byla 1.4. V této době se jednalo o stabilní verzi. Asterisk byl však v průběhu práce aktualizován na poslední stabilní verzi 1.6. Tato aktualizace přinesla mnoho problémů a to především systémům třetích stran pro vyúčtování. Aktualizace měla pro mě osobní přínos, protože jsem v té době poznal hlavní rozdíly v konfiguraci verze 1.4 a verze 1.6. Problémy s aktualizací byly vyřešeny a to především díky logům.

Po instalaci Asterisku byly postupně zkoušeny možnosti ukládání CDR záznamů. Jako první možnost bylo vyzkoušeno ukládání do textových souborů ve formátu csv. Tato cesta se ukázala jako slepé řešení, které lze velmi těžko rozšířit. Existují však skripty, které tyto textové záznamy přeformulují tak, aby bylo možné je importovat do databáze. Pokud tedy na nějaké starší VoIP ústředně jsou tyto záznamy v textovém formátu, je možné je importovat do databáze a dále s nimi pracovat.

Druhou vyzkoušenou možností bylo ukládání CDR záznamů přímo do databáze MySQL. Tato možnost se ukázala jako velmi efektivní, protože bylo možné oddělit účtovací systém od systému Asterisk. Není tedy nutností mít vše na jednom serveru. Další výhodou byla jednoduchá vizualizace dat přes skriptovací jazyky. V návaznosti na tuto možnost byla vytvořena vlastní aplikace v jazyce PHP, která dokáže přehledně zobrazit CDR záznamy o hovorech. Dokáže třídit tyto záznamy podle zadaných kritérií a také vypočítat dobu a cenu hovorů. Výsledná aplikace je tak vhodným nástrojem a doplňkem pobočkových ústředen. Pokud majitel pobočkové ústředny nemá online přehled nad výdaji za telefonní služby, má k dispozici pouze měsíční účty od poskytovatele VoIP služeb, tak aplikace je v tomto případě vhodné řešení pro online náhled a kontrolu pobočkové ústředny.

Další vyzkoušenou variantou pro zobrazení záznamů o volání byl systém A2billing. Tento systém má velký přínos především ve vizualizaci CDR záznamů. Možnosti tohoto systému jsou předurčeny především pro VoIP poskytovatele. A2billing je sice open source program, ale jeho dokumentace je natolik nedostatečná, že jakákoli implementace pro poskytovatele služeb bude vyžadovat asistenci a placenou podporu.

V poslední kapitole jsem se věnoval bezpečnosti VoIP ústředen a to především využití CDR záznamů pro odhalení zneužití služeb. Poskytovatelé VoIP služeb většinou nemají problém s implementováním limitů za služby hovorného. Pokud účastník dosáhne tohoto limitu, tak další hovory jsou mu znemožněny. Také blokování volání na drahé telefonní linky je dnes možné. Poskytovatelé v některých případech tyto hovory blokují. Rozhodnutí, jestli si nastavit limit za služby hovorného nebo jestli si nechat zablokovat volání na drahá telefonní čísla, je vždy na zákazníkově. Chybí však možnost kontroly a upozornění zákazníka ze strany VoIP poskytovatele v případě, že zákazník vede dlouhé hovory například na drahé zahraniční telefonní linky.

Dalším přínosem této práce je tedy skript, který ukazuje třetí možnou cestu obrany proti zneužití VoIP ústředny. V diplomové práci jsem implementoval skript, který v pravidelných intervalech vypočítává statistiky délky hovorů u všech uživatelů, tato data porovnává s limity a v případě podezřelého chování uživatelů informuje o situaci administrátora. V diplomové práci není pouze vyřešen problém účtovacího systému, ale také je uděláno řešení, které staví účtovací systém do pomocníka v otázce bezpečnosti.

Literatura

- [1] Instalační příručka systému Debian. Debian, 2011 Dostupný z <http://www.debian.org/releases/stable/i386/index.html.cs>
- [2] Instalace webového serveru apache2 s podporou php5 a MySQL, 2011. Dostupný z <http://www.howtoforge.com/installing-apache2-with-php5-and-mysql-support-on-debian-lenny-lamp>
- [3] OPPEL, Andrew. Databáze bez předchozích znalostí: průvodce pro samouky. Jiří Matoušek; David Krásenský. 1. vyd. Brno: Computer Press, 2006. 320 s. ISBN 80-251-1199-7.
- [4] apache http server documentation, apache software foundation, 2011 Dostupný z <http://httpd.apache.org/docs/>
- [5] VOZŇÁK, Miroslav. Voice over IP. Ostrava: VŠB – TU Ostrava, 1. vydání, 2008. 176 s. ISBN 978-80-248-1828-3.
- [6] MEGGELEN, J., MADSEN, L., KIELHOFNER, K. Asterisk Cookbook. O'Reillz Media, 2007. ISBN 978-0596526924
- [7] VoIP wiki. 2011 Dostupný z <http://www.voip-info.org/>
- [8] V. MEGGELEN, Jim, SMITH, Jared, MADSEN, Leif. Asterisk: The Future of Telephony. Mike Loikides; Colleen Gorman. 1st edition. Sebastopol : O'Reilly Media, 2005. 376 s. Dostupné z: <http://www.cyberciti.biz/tips/download-asterisk-howto-tutorialbook.html> ISBN 0-596-00962-3.
- [9] Kosek Jiří, Tvorba internetových aplikací, 1999, 492 s. Dostupný z <http://www.kosek.cz/php/php-tvorba-interaktivnich-internetovych-aplikaci.pdf> , ISBN 80-7169-373-1